Indirection

*Indirection*: rather than reference an entity directly, reference it (“indirectly”) via another entity, which in turn can or will access the original entity

“Every problem in computer science can be solved by adding another level of indirection"

-- Butler Lampson
Multicast: Act of sending datagram to multiple receivers with single “transmit” operation

Analogy: One teacher to many students

Question: How to achieve multicast

Network multicast

Router actively participate in multicast, making copies of packets as needed and forwarding towards multicast receivers
multicast group concept: use of indirection

- hosts addresses IP datagram to multicast group
- routers forward multicast datagrams to hosts that have “joined” that multicast group
Multicast groups

- Class D Internet addresses reserved for multicast:
  - 1110 | Multicast Group ID
  - 28 bits

- Host group semantics:
  - anyone can “join” (receive) multicast group
  - anyone can send to multicast group
  - no network-layer identification to hosts of members

- **Needed:** Infrastructure to deliver mcast-addressed datagrams to all hosts that have joined that multicast group
Joining a mcast group: Two-step process

- **Local:** Host informs local mcast router of desire to join group: IGMP (Internet Group Management Protocol)
- **Wide area:** Local router interacts with other routers to receive mcast datagram flow
  - many protocols (e.g., DVMRP, MOSPF, PIM)
Multicast via Indirection: Why?

- Don't need to individually address each member in the group: header savings
- Looks like unicast; application interface is simple, single group
- Abstraction, delegating works of implementation to the routers
- More scalable because, sender doesn't manage the group, as receivers are added, new receivers must do the work to add themselves
Mobility and Indirection

How do you contact a mobile friend?

Consider friend frequently changing addresses, how do you find her?

- Search all phone books?
- Call her parents?
- Expect her to let you know where he/she is?

I wonder where Alice moved to?
Mobility and indirection

- Mobile node moves from network to network
- Correspondents want to send packets to mobile node
- Two approaches:
  - \textit{Indirect routing}: Communication from correspondent to mobile goes through home agent, then forwarded to remote
  - \textit{Direct routing}: Correspondent gets foreign address of mobile, sends directly to mobile
**Mobility: Vocabulary**

*Home network:* permanent "home" of mobile (e.g., 128.119.40/24)

*Permanent address:* address in home network, *can always* be used to reach mobile (e.g., 128.119.40.186)

*Home agent:* entity that will perform mobility functions on behalf of mobile, when mobile is remote

*Wide area network*

*Correspondent*
Mobility: More vocabulary

**Permanent address:** remains constant (e.g., 128.119.40.186)

**Visited network:** network in which mobile currently resides (e.g., 79.129.13/24)

**Care-of-address:** address in visited network. (e.g., 79,129.13.2)

**Foreign agent:** entity in visited network that performs mobility functions on behalf of mobile.

**Correspondent:** wants to communicate with mobile.
Mobility: Registration

End result:
- Foreign agent knows about mobile
- Home agent knows location of mobile

1. Foreign agent contacts home agent home: “this mobile is resident in my network”
2. Wide area network

home network

visited network

mobile contacts foreign agent on entering visited network
Mobility via Indirect Routing

1. Correspondent addresses packets using home address of mobile.
2. Home agent intercepts packets, forwards to foreign agent.
3. Foreign agent receives packets, forwards to mobile.
4. Mobile replies directly to correspondent.

Wide area network
Indirect Routing: Comments

- Mobile uses two addresses:
  - **Permanent address**: used by correspondent (hence mobile location is *transparent* to correspondent)
  - **Care-of-address**: used by home agent to forward datagrams to mobile
- Foreign agent functions may be done by mobile itself
- **Triangle routing**: correspondent-home-network-mobile
  - Inefficient when correspondent, mobile are in same network
Indirect Routing: Moving between networks

- Suppose mobile user moves to another network
  - Registers with new foreign agent
  - New foreign agent registers with home agent
  - Home agent update care-of-address for mobile
  - Packets continue to be forwarded to mobile (but with new care-of-address)

- Mobility, changing foreign networks transparent: **Ongoing connections can be maintained!**
Mobility via Direct Routing

1. Correspondent requests, receives foreign address of mobile.
2. Correspondent forwards to foreign agent.
3. Foreign agent receives packets, forwards to mobile.
4. Mobile replies directly to correspondent.
Mobility via Direct Routing: Comments

- Overcome triangle routing problem
- Non-transparent to correspondent:
  Correspondent must get care-of-address from home agent
  - What happens if mobile changes networks?
Mobile IP

- RFC 3220
- Has many features we’ve seen:
  - home agents, foreign agents, foreign-agent registration, care-of-addresses, encapsulation (packet-within-a-packet)
- 3 components to standard:
  - agent discovery
  - registration with home agent
  - indirect routing of datagrams
Mobility via indirection: Why indirection?

- Transparency to correspondent
- “ Mostly” transparent to mobile (except that mobile must register with foreign agent)
  - transparent to routers, rest of infrastructure
  - potential concerns if egress filtering is in place in origin networks (since source IP address of mobile is its home address): spoofing?
Content Delivery Networks: Indirection with DNS
Internet Content

- Content is
  - static web pages and documents
  - images and videos, streaming, ...

- Content becomes more and more important!
  - 500 exabytes ($10^{18}$) created in 2008 alone [Jacobson]
  - Estimated inter-domain traffic rate: 39.8 TB/s [Labovitz]
  - Annual growth rate of Internet traffic: ~40%-60% [Labovitz]
  - Much of web growth due to video (Flash, RTSP, RTP, YouTube, etc.)

- How to deliver content?
- How to cope with growth of content?

Following slides adapted from Wolfgang Mühlbauer
Application mix

Application mix in 2009

- HTTP dominates

Inside HTTP
- Flash-video dominates
- Images and RAR files next

Prevalence of CDNs

- 30 (out of ~30000) ASes contribute 30% of inter-domain traffic
- July 2009: CDNs originate at least 10% of all inter-domain traffic
- Top ten origin ASes in terms of traffic

<table>
<thead>
<tr>
<th>Rank</th>
<th>Provider</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Google</td>
<td>5.03</td>
</tr>
<tr>
<td>2</td>
<td>ISP A</td>
<td>1.78</td>
</tr>
<tr>
<td>3</td>
<td>LimeLight</td>
<td>1.52</td>
</tr>
<tr>
<td>4</td>
<td>Akamai</td>
<td>1.16</td>
</tr>
<tr>
<td>5</td>
<td>Microsoft</td>
<td>0.94</td>
</tr>
<tr>
<td>6</td>
<td>Carpathia Hosting</td>
<td>0.82</td>
</tr>
<tr>
<td>7</td>
<td>ISP G</td>
<td>0.77</td>
</tr>
<tr>
<td>8</td>
<td>LeaseWeb</td>
<td>0.74</td>
</tr>
<tr>
<td>9</td>
<td>ISP C</td>
<td>0.73</td>
</tr>
<tr>
<td>10</td>
<td>ISP B</td>
<td>0.70</td>
</tr>
</tbody>
</table>

Source: Craig Labovitz et al. Internet Inter-Domain Traffic. SIGCOMM 2010.
Why not Serving Content from One’s Own Site?

- Enormous demand for popular content
  - Cannot be served from single server
- Bad performance
  - Due to large distance: TCP-throughput depends on RTT!
  - Bad connectivity?
- Single point of “failure”
  - High demand leads to crashes or high response times (e.g., flash crowds)
- High costs
  - Bandwidth and disk space to serve large volumes (e.g., videos)
Approaches to Content Delivery

- **Idea**: replicate content and serve it locally

- **Content distribution networks (CDN)**
  - Offload content delivery to large number of content servers
  - Put content servers near end-users

- **Peer-to-peer networks**
  - In theory: infinite scalability
  - Yet: download capacity throttled by uplink capacity of end users
Akamai – A Large CDN

- Akamai (Hawaiian: “intelligent”)
  - Evolved out of MIT research effort: handle flash crowds
  - > 70000 Servers located in 72 countries, > 1000 ASs
  - Customers: Yahoo!, Airbus, Audi, BMW, Apple, Microsoft, etc.

- Why using Akamai?
  - *Content consumer*: Fast download
  - *Content provider*: Reduce infrastructure cost, quick and easy deployment of network services

- Task of CDNs: Serve content
  - *Static web content*: HTML pages, embedded images, binaries ...
  - *Dynamic content*: break page into fragments; assemble on Akamai server, fetch only noncacheable content from origin website:
  - *Applications*: audio and video streaming
Akamai: Is the Idea Really That Novel?

- Local server cluster
  - Bad if data center or upstream ISP fails

- Mirroring
  - Deploying clusters in a few locations
  - Each mirror must be able to carry all the load

- Multihoming
  - Using multiple ISPs to connect to the Internet
  - Each connection must be able to carry all the load

- Akamai vastly increases footprint
  - monitors and controls their worldwide distributed servers
  - directs user requests to appropriate servers
  - handles failures
Akamai Relies on DNS Redirection

- Example: Access of Apple webpage (www.apple.com)
- Pictures are hosted by Akamai: images.apple.com
- Type: `dig images.apple.com` into your Linux shell

```
[...]
;; ANSWER SECTION:
[more CNAME redirections]
images.apple.com.edgesuite.net.globalredir.akadns.net. 2961 IN      CNAME    a199.gi3.akamai.net.
a199.gi3.akamai.net. 10 IN       A       184.84.182.56
a199.gi3.akamai.net. 10 IN       A       184.84.182.66
[...]
```

DNS redirects request to DNS servers controlled by Akamai!
Akamai Deployment

- Edge server organized as “content cluster”
  - in many Autonomous Systems
  - multiple servers
  - local “low-level” DNS server

Client directed to “closest” server
How does Akamai work? (simplified)

Normal web request:
- First DNS resolution
- Then HTTP connection

Slide adapted from “Drafting Behind Akamai”, Sigcomm 2006
How does Akamai work? (simplified)

DNS request for "Akamized" content:
- Results in CNAME

Slide adapted from “Drafting Behind Akamai”, Sigcomm 2006
How does Akamai work? (simplified)

Slide adapted from "Drafting Behind Akamai", Sigcomm 2006
How does Akamai work? (simplified)

Slide adapted from "Drafting Behind Akamai", Sigcomm 2006
Two-level server assignment

- Akamai top-level DNS server
  - Anycasted
  - Selects location of “best” content cluster
  - Delegates to content cluster’s low-level name server
  - TTL 1 hour

- Akamai low-level server
  - Return IP addresses of servers that can satisfy the request: consistent hashing
  - TTL 20 seconds: quick adoption to load conditions

- Most CDNs use similar techniques
  - Some CDNs rely on Anycast to send traffic to closest content server (e.g., Limelight)
What is the „best“ location?

- Service requested
  - Server must be able to satisfy the request (e.g., QuickTime stream)

- Server health
  - Up and running without errors

- Server load
  - Server’s CPU, disk, and network utilization

- Network condition
  - Minimal packet loss to client, sufficient bandwidth to handle requests

- Client location
  - Server should be close to client, e.g., in terms of RTT
Indirection: Summary

We’ve seen indirection used in many ways:
- Multicast
- Mobility
- CDNs

The uses of indirection:
- Sender does not need to know receiver id – do not want sender to know intermediary identities
- Load balancing
- Beauty, grace, elegance
- Transparency of indirection is important
- Performance: Is it more efficient?